Amazon, CN

Technical Document

A Transportation Efficiency Optimization Algorithm

Author:
Luping Zhao

February 22, 2018

Contents

1 Introduction

Between the Vendor and the Fulfillment Center(FC) of Amazon, there are two kinds of logistics—the forward logistics and reverse logistics, and they are named as AITS and V-Return, respectively.
In the AITS, the Amazon first order certain goods from the vendor, within 48 hours, the Vendor will pack ordered goods awaiting to be picked up by vehicles from Fulfillment Centers of Amazon. According to contracts, there are only 24 hours for Amazon to pick up the goods. normally, these orders are sent on Monday and Wednesday, which directly leads to an uneven distribution of the transportation traffic among a week, the traffic happens on Wednesday and Friday account for 60% of total traffic.
In contrast with the AITS, V-Return is a service designed to return goods that are no longer needed. Note that this service's time limit is less strict than AITS, normally within a week, and its required traffic is much lower, as a result V-Return uses the same vehicles on Monday and Tuesday to avoid conflicts with AITS.
The orders in these two services are arranged manually, only a small fraction of goods in these services are transported together, leading to potential space for optimization. That is why I propose an optimization method to merge a fraction of traffic in AITS and V-Return to generate a more efficient solution.
Currently, the AITS traffic solution is formed by long-term evolution, we hence focus on adjustment of the orders in V-Return in following aspects:

- Screen qualified orders in V-Return and merge them into AITS traffic, aiming to reduce total travel distance theoretically.
- Verify the optimization solution by Amazon's potential opportunity analysis, validate parameters such as vehicles' loading rate.

1.1 Methodology

Using Python to develop the main functional module, it also functions as a sub-module of existing information system.

- Input: Retrieve data of AITS and V-Return, save them as local files.
  - AITS manual traffic scheme.

  - V-Return orders.

- Calculating optimized traffic scheme.
- Output: Traffic scheme after adjustment (Summarized as plate numbers)
  - Specific V-Return delivery recommendations
  - Insert position and amount of V-Return orders
  - Analyze total travel distance before and after the adjustment

2 Model logic and algorithm

The flow chart of the algorithm is showed in 1.

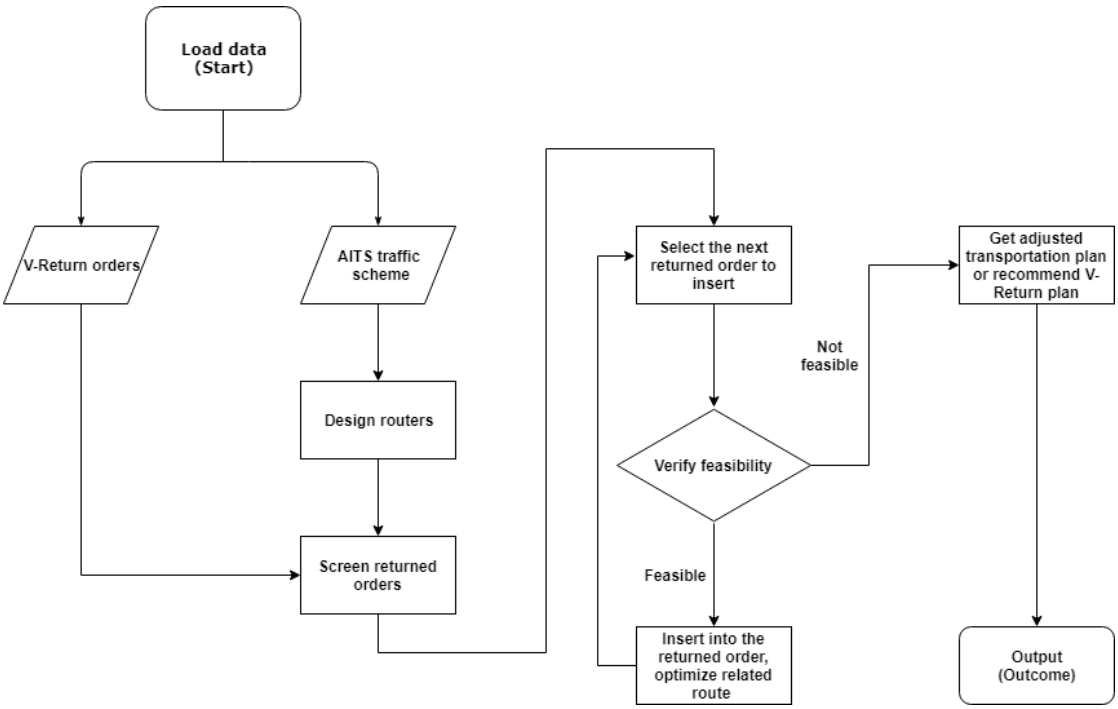

Figure 1: Flow chart of the algorithm

The algorithm is consisted by four main functional module: prior screen of returned order for potential candidates; Select the next return order to insert; feasibility verification and route optimization.

2.1 V-Return order screening

The set of candidates for the merge is selected among the V-Return nodes that are within a certain distance of AITS nodes, when any of the following conditions are satisfied, we terminate the screening and the candidate set is generated:

- All V-return nodes within 5 km of AITS nodes of the current route are selected into candidate set.
- The sum of AITS nodes of current route and the nodes in candidate set is over 15.

All adjustments are limited to the candidate set, other nodes are considered in the next round of screening.

2.2 V-Return insertion assessment

There are two criterions in V-Return order insertions, the additional travel distance needed and the quantity of the cargo transported.

- Additional travel distance: if we insert a point k between points i and j, the additional travel distance is defined by

  $$c_d = d_{i,k} + d_{k,j} - d_{i,j}$$

  where $d_{i,j}$ denotes the distance between point i and point j, intuitively, we expect the additional travel distance to be minimized.

- After discussing with colleagues in fullfillment centers, generally we think larger cargo quantity is better. After rescaling the cargo quantity, we have $c_c = f_k \cdot L/C$, where f_k is the cargo quantity, L denotes the total distance of current route and C denotes the capacity of vehicles.

We take the linear combination of the above two standards and create a index for general assessment:

$$c = c_d - \lambda c_c$$

where the candidate is selected by minimizing index c, plus the feasibility requirements are met. λ denotes the weight we put on cargo capacity, a larger λ means we value cargo capacity more, a smaller λ means we value the effect of additional travel distance. For simplicity, we choose here $\lambda = 1$.

2.3 Feasibility verification

We mainly verify the following aspects:

- Total time en route: To guarantee the cargo of AITS's timely delivery, the cargo needs to reach destinations within certain time limit.
  - Time limit: 7 hours (8:30-15:30)
  - The time of traveling from the first vendor to fulfillment center.
  - Influencing Factors: time en route and time of service.
    - Time en route: calculated based on the cargo quantity of a point, less than 10 crates-15 min, and 30s for one additional crate over 10. We also add 10% of service time for the additional moving needed.
    - service time: 1.4 multiplies the time estimated by navigation API.
- Cargo quantity:
  - Verify the cargo quantity at each stop.
  - Consider both the volume and the weight.
- Quantity of crates: The loading cost is calculated by stepwise pricing, every 400 crates for a worker. We can consider a marginal increase in loading cost as a signal of invalid options(for example, a certain order, say, 5 crates, increase the crates quantity from 397 to 402, which makes the order need one additional loader, such a marginal increase is regarded as not feasible.).

2.4 TSP algorithm

The determination of the route of AITS and the ensuing optimization all need the calculation of TSP, the model is based on a 3-opt algorithm to optimize the route. First, divide the entire route into three segments, then delete their connections and reconnect the three segments in all other possible ways, and then evaluating each reconnection method to find the optimum one. This process is then repeated for a different set of 3 connections. The outcome can be seen as a neighborhood as previous outcomes, and the process stops until no better outcome exists in the neighborhood. The process is illustrated in 2.

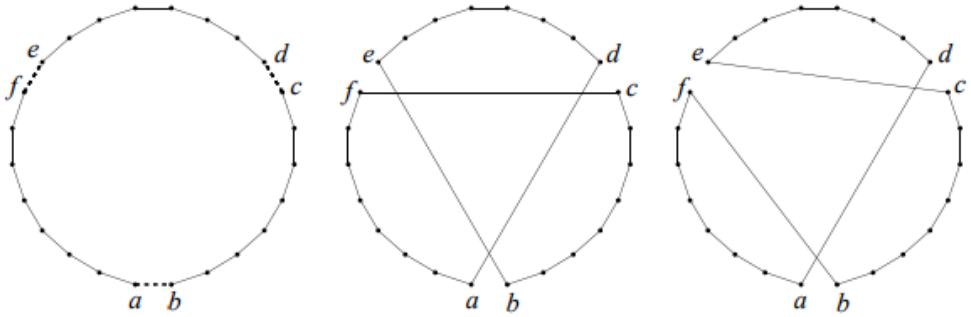

Figure 2: 3-opt algorithm

When we design AITS routes, first we generate an initial solution randomly, and then optimize it using 3-opt algorithm.

2.5 Example

The figure 3 is an example of this process, the respective meaning of these plots are:

1 Load AITS data.

2 Design AITS routes.

3 Screen to get a V-Return order candidate set.

4,6,8 Choosing the V-Return order to insert, verify feasibility.

5,7 If feasible, adjust the route.

9 Not feasible, insertion ends.

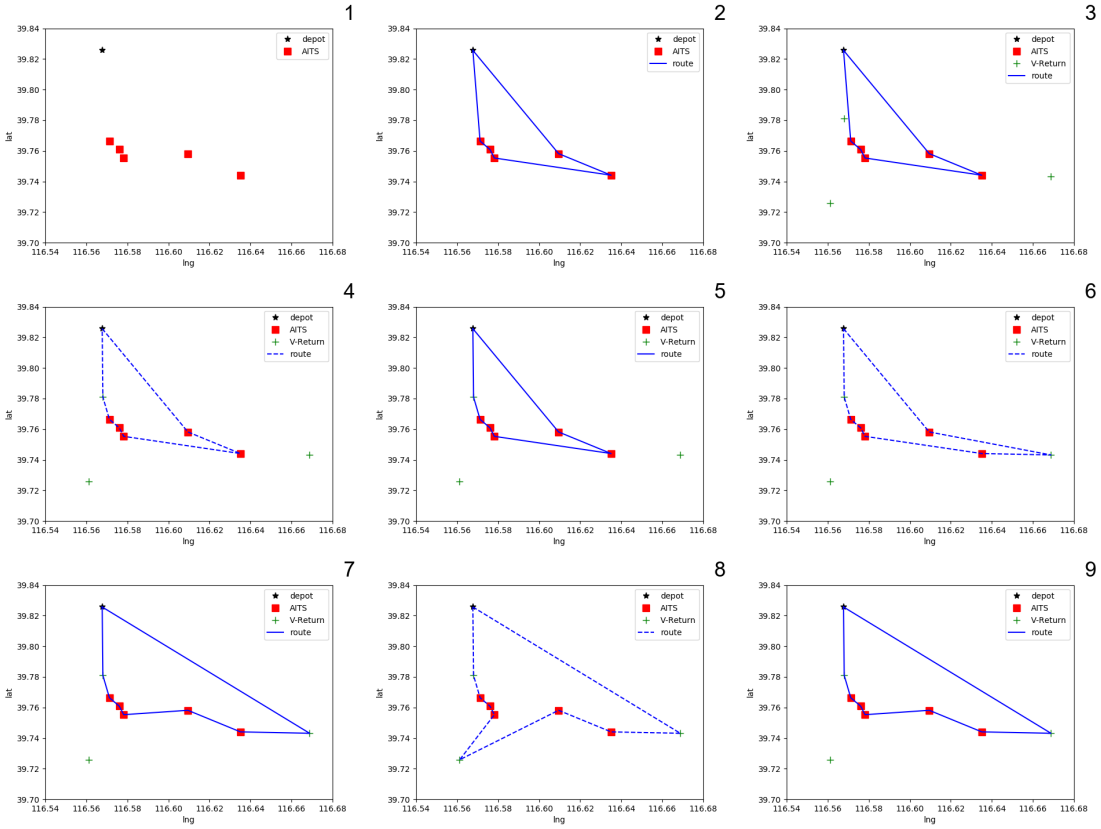

Figure 3: An example of optimization process

3 Historical data backtesting

3.1 Data from the second half of 2016

The datasets include records of vehicle sending of AITS and V-Return. The data is divided by day, we expect an adjusted route as the model's output, in the case of no recommendations, we choose to stick with the original V-Return transportation plan (use reserved vehicles for V-Return) or reassign the V-Return vehicles that travel the least positions.

There are two main factors to consider in backtesting: the quantity of cargo inserted (in crates), and the total travel distance, by travel distance we mean the theoretical distance retrieved from Google maps and other third-party APIs. The process of distance determination before and after using the model are shown in

the following table:

Before using our new model	Optimize every route of AITS and V-Return separately.
After using our new model	Focusing on the adjusted route (for delivery of original AITS cargo and the newly inserted V-Return cargo), solve the Traveling Salesman Problem. Other V-Return orders are delivered by the original route, solve related TSP.

Table 1: Difference of distance optimization before and after using the new model

In terms of cargo quantity, the historical data shows that approximately 15% of V-Return cargo were delivered with AITS cargo, after optimization, approximately 50% of V-Return cargo are delivered with AITS. Total travel distance is also reduced, about 2% on average.

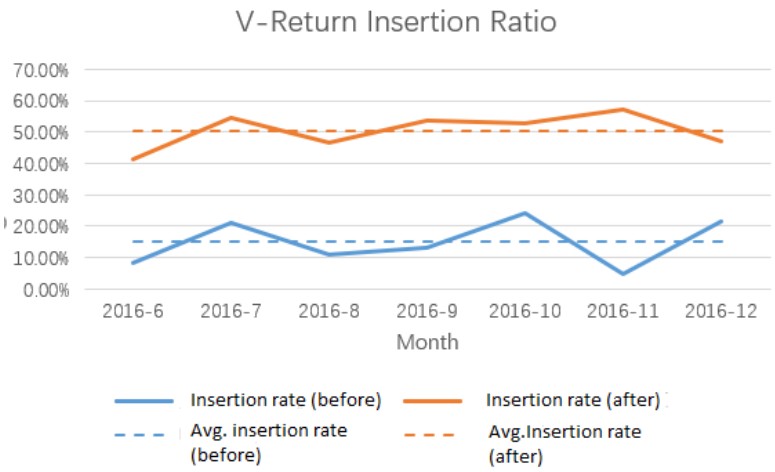

Figure 4: Insertion rate optimization outcome (2016)

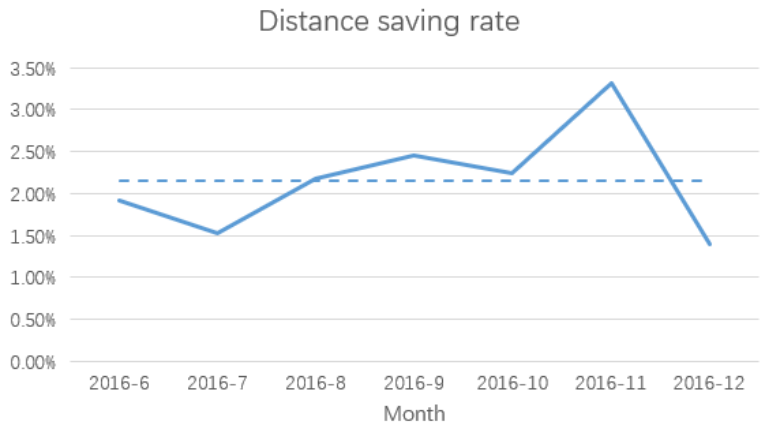

Figure 5: Total distance optimization outcome (2016)

3.2 Data from June and July of 2017

Using the same method, we get the following outcome with new datasets(the dotted line indicates average insertion rate of the second half of 2016.).

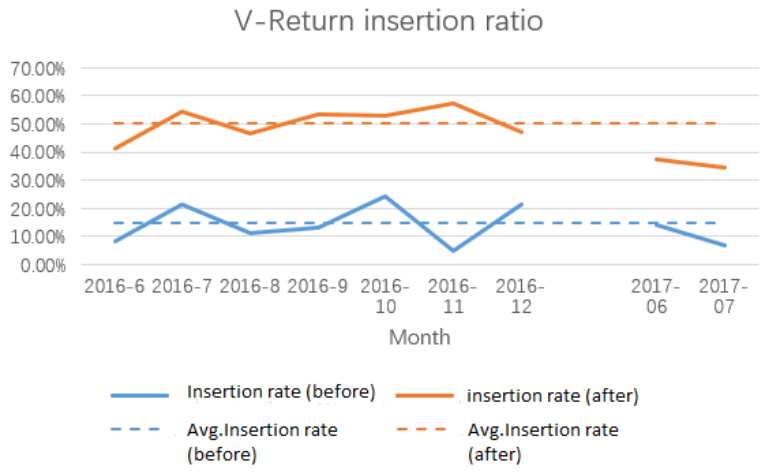

Figure 6: Insertion rate optimization outcome (2017)

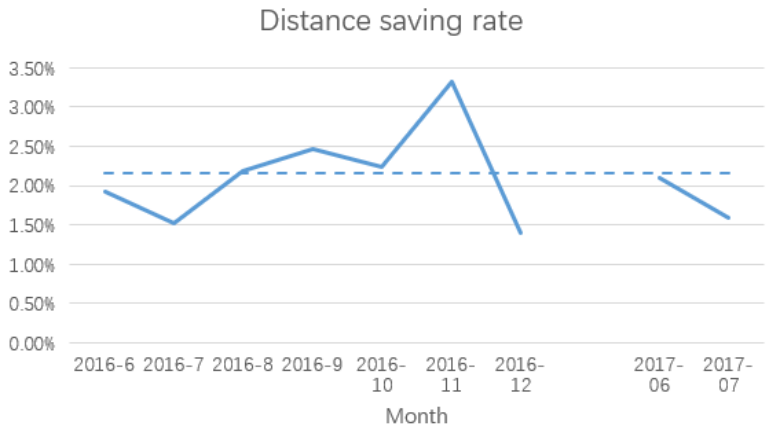

Figure 7: Total distance optimization outcome (2017)

Due to limited observations, our analysis is also limited, but we can still see that there were approximately 36% of V-Return cargo that can be transported with AITS cargo, and total travel distance is reduced by 1.85%. the outcome is less ideal than the outcome of 2016, there are two possible explanations:

- Reduced V-Return cargo quantity (460 crates/day→390 crates/day).
- Increased AITS nodes and loading rate (4.3 nodes/vehicle→6.2 nodes/vehicle;60%→%85).

3.3 Other insertion patterns

The backtesting so far is limited within each day. In reality, V-Return orders has a rather loose schedule, so we can insert qualified V-Return orders into AITS orders at another day, in other words, we consider insertions starting from the day the V-Return orders generate (instead of the day of scheduled delivery), and the insertions are no longer limited to AITS orders within the same day. However, weekends are special cases, on weekends, we do not insert orders from weekdays. Our new insertion patterns consider the orders that qualify following conditions:

- Original V-Return orders that can be delivered in advance.
- V-Return orders that are generated on a day but are scheduled to be delivered afterwards.

3.3.1 Divided by week

Using data from 2017, our new insertion pattern shows that 59.8% of V-Return cargo can be delivered with AITS, and the total travel distance saving rate is 3.7%, that means almost 100 kilometers per day.

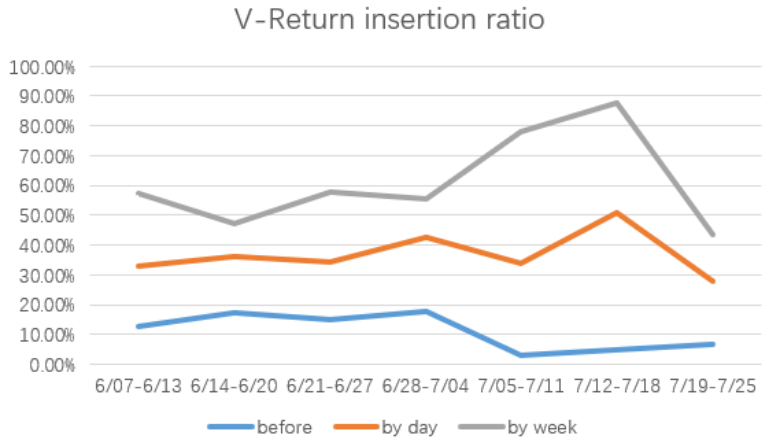

Figure 8: Insertion rate optimization outcome (by week, 2017)

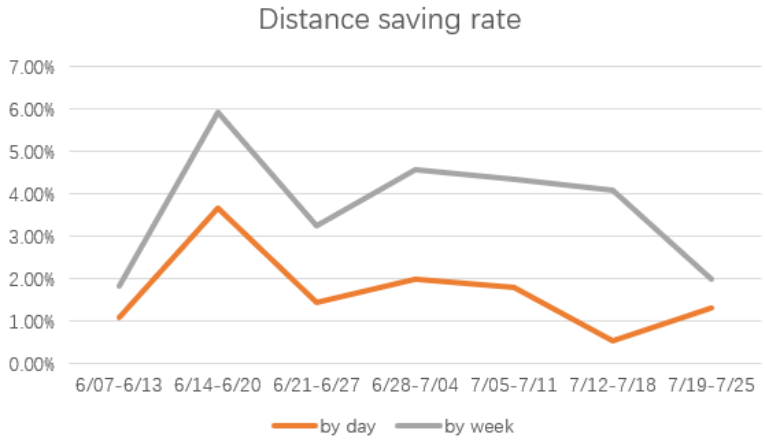

Figure 9: Total distance optimization outcome (by week, 2017)

3.3.2 Divided by month

The time limit of V-Return orders is one-week, however, when we divide the orders by week, there will be uneven distribution of V-Return orders, to eliminate this effect, we divide the orders by month without lifting time restriction. The outcome shows that 52.5% of V-Return cargo can be delivered with AITS, and the total travel distance saving rate is 3.5%.

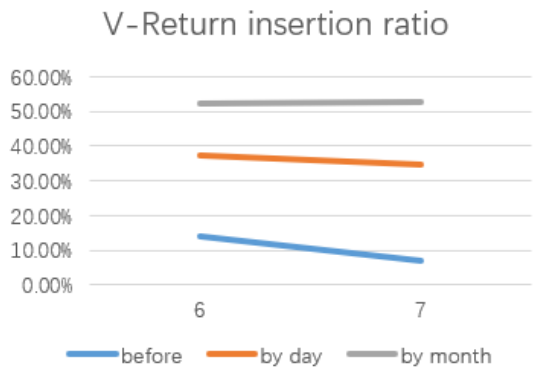

Figure 10: Insertion rate optimization outcome (by month, 2017)

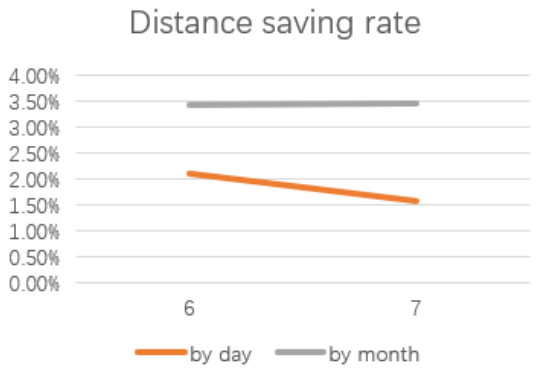

Figure 11: Total distance optimization outcome (by month, 2017)

3.4 Assessment of nodes and loading rate

Based on outcomes in 3.3.2, in June and July of 2017, AITS delivered 1152 orders, 418 of them are inserted with V-Return orders. In these 418 orders, 3.03 extra

nodes were added and the loading rate of V-Return cargo is 20.1% by weight and 7.68% by volume, all on average.

3.5 Cost analysis

Based on outcomes in 3.3.2, we can calculate the cost savings, the outcomes are:

Gasoline Saving	Loading Fee Saving	Total Saving (Jun & Jul)	Monthly Saving
5,141.39	1,891.89	7,033.29	3,516.64

Table 2: Cost savings

A Technical instructions of the compiled tool

A.1 File directory

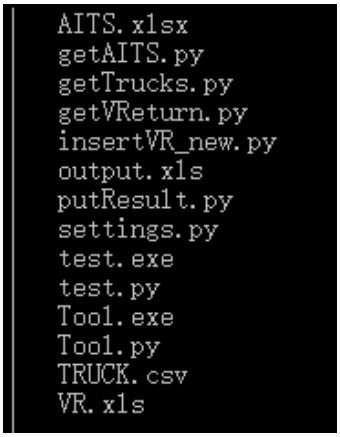

Figure 12: File directory

The directory should include the following files:

- AITS.xlsx/VR.xls: AITS vehicle sending record and V-Return order datasets.
- TRUCK.csv: Truck info.
- output.xls: Output

- getAITS.py/getVR.py/getTruck.py: data retrieve
- insertVR_new.py: route adjustment
- putResult.py: gnerate output
- settings.py: parameters settings
- test.py/test.exe: compiled tool for command line
- Tool.py/Tool.exe: GUI

A.2 Direct usage

If you do not want any modifications to the parameters, you can use the executable file without any extra installations.

A.2.1 GUI

Click Tool.exe, select the AITS and VR files, click start, the output should be a output.xls file in the same directory as the tool.

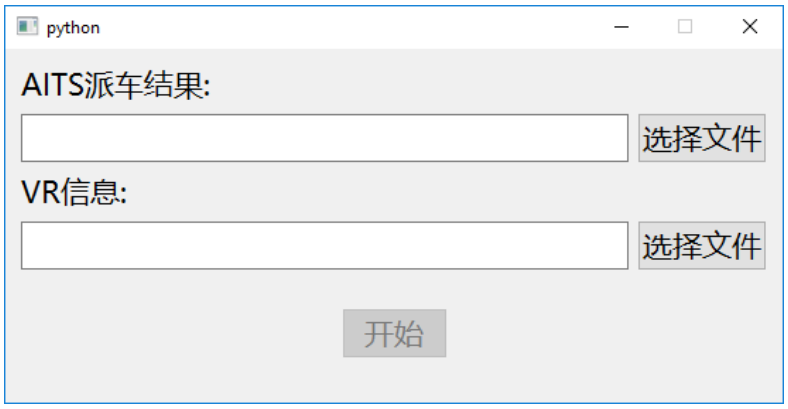

Figure 13: GUI

A.2.2 Command line calling

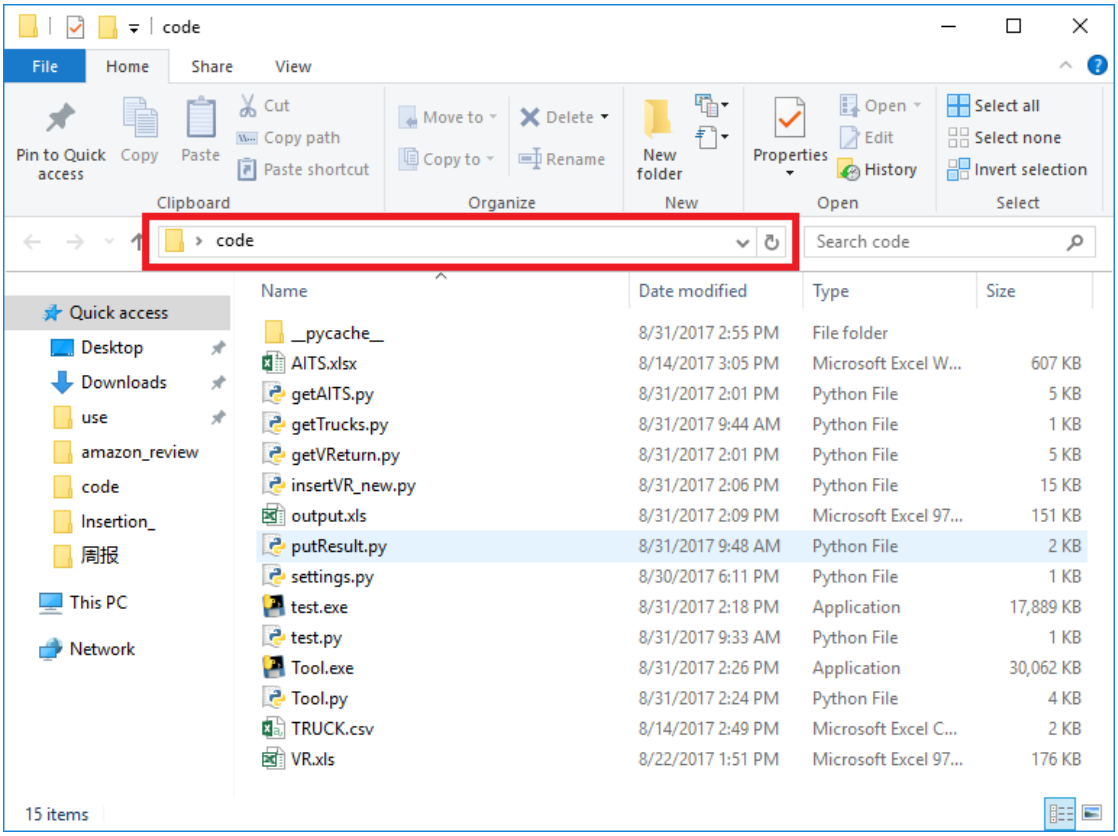

Figure 14: CMD

Use CMD (type CMD in the red area in 14) and enter the following code:

```
test.exe [AITS file name] [VR file name]
```

A.2.3 VB calling

Within the same directory, use the following code for VB:

```
Private Sub abc()
Path = ThisWorkbook.Path
Disc = Left(Path, 2)

s = "cmd /k cd " & """" & Path & """"
p1 = "&& cmd /k " & "test.exe "
```

```
p2 = "AITS.xlsx "
p3 = "VR.xls"


Shell s & p1 & p2 & p3, vbNormalFocus
End Sub
```

A.3 Python environment

You will need Python 3 and the following packages, you can install them by using pip:

```
pip install xlrd
pip install xlwt
pip install numpy
pip install pyqt5
pip install pyinstaller
```

A.4 Parameter modifications

You can change the parameters in the file settings.py.

Name	**Description**
key	The key for using third-party navigator API
size	Upper bond of AITS and V-Return nodes
radius_bound	Upper bond for distance of V-Return candidate from nearest AITS node
lam	Weight parameter
upperbound	Total time limit
calcServiceTime(carton)	Logic of crates approximity
VR_allowance	Amplifier for service time of V-Return
TT_allowance	Amplifier for time en route

Table 3: Parameter modifications